What is claimed is:

Claims

1. A method for collecting information on components in an information technology (IT) system, comprising:

discovering the existence of at least one of the components in the IT system;

determining at least one dependency between two or more of the components; and

tracking changes to at least one of the components and the dependency between two or more of the components.

2. The method of claim 1, further comprising generating a visual map of the IT system, the visual map including a depiction of at least one of the components and the at least one dependency between two or more of the components.

3. The method of claim 2, wherein the visual map includes tracked changes to at least one of the components.

4. The method of claim 1, wherein at least one of the components is an application.

5. The method of claim 1, wherein discovering the existence of at least one of the components includes:

receiving event information regarding an occurrence in the IT system, the occurrence relating to a first component;

comparing the first component along with other components to at least one fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component; and

determining that at least one of the components exists when all of the elements of the fingerprint corresponding to the known component are matched.

6. The method of claim 5, wherein the occurrence is selected from one or more of a file creation, a file deletion, and a file modification.

7. The method of claim 5, wherein the occurrence is selected from one or more of a registry key creation, a registry key deletion, and a registry key modification.

8. The method of claim 5, wherein the occurrence is information regarding detection of a particular component in the IT system.

9. The method of claim 5, further comprising indicating that a particular component has been damaged if the occurrence is a deletion and at least one of the elements of the fingerprint are no longer matched by the components in the IT system.

10. The method of claim 5, further comprising indicating that a particular component has been uninstalled if the occurrence is a deletion and all of the elements of a minimum set of the fingerprint are no longer matched by the components in the IT system.

11. The method of claim 1, wherein the at least one dependency is selected from the group consisting of shared library usage, network usage, and containment dependencies.

12. The method of claim 1, further comprising:

generating a component discovered message upon the discovery of one of the components;

retrieving a list of elements to track for the discovered component; and

using the list of elements to track changes to the discovered component.

13. An agent for collecting information on components in an information technology (IT) system, the agent residing on a computer in the IT system, the agent comprising:

an observer module to detect event information about elements of the computer; and

an analysis module to process the event information, the analysis module including: (a) component discovery rules to process event information and match event information with elements of one or more fingerprints of known components using an accumulator to discover the existence on the IT system of at least one of the components, and (b) dependency discovery rules to detect relationships between components of the IT system.

14. A system for collecting information on components in an information technology (IT) system, comprising:

means for discovering the existence of at least one of the components in the IT system;

means for determining at least one dependency between two or more of the components; and

means for tracking changes to at least one of the components and the dependency between two or more of the components.

15. An apparatus for collecting information on components in an information technology (IT) system, comprising:

a memory storing a program;

a processor in communication with the memory; in which the processor is directed by the program to:

discover the existence of at least one of the components in the IT system;

determine at least one dependency between two or more of the components; and

track changes to at least one of the components and the dependency between two or more of the components.

16. A method for discovering components in an information technology (IT) system, comprising:

receiving event information regarding an occurrence in the IT system, the occurrence relating to a first component;

comparing the first component along with other components to at least one fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component; and

if the first component and the other discovered components match substantially all of the key low-level elements of the fingerprint, using a subfingerprint of a known refined component to discover the existence of a second component that corresponds to the known refined component.

17. The method of claim 16, wherein the known refined component is a version of the known component.

18. The method of claim 16, wherein the known refined component is an optional piece of the known component.

19. The method of claim 16, further comprising generating a command message to collect further information if all of the low-level elements of the fingerprint are matched.

20. The method of claim 19, further comprising receiving event information in response to the command message, wherein the event information is used with the subfingerprint of the known refined component to discover the existence of the second component.

21. The method of claim 16, further comprising detecting low-level items in the IT systems and generating event information regarding the low-level items.

22. The method of claim 21, wherein the low-level items are selected from one or more of files, registry settings, and database schemas.

23. A computer-readable medium for discovering components in an information technology (IT) system, the computer-readable medium storing instructions that direct a microprocessor to:

receive event information regarding an occurrence in the IT system, the occurrence relating to a first component;

compare the first component along with other components to at least one fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component; and

if the first component and the other discovered components match substantially all of the key low-level elements of the fingerprint, use a subfingerprint of a known refined component to discover the existence of a second component that corresponds to the known refined component.

24. An apparatus for discovering components in an information technology (IT) system, comprising:

a memory storing a program;

a processor in communication with the memory; in which the processor is directed by the program to:

receive event information regarding an occurrence in the IT system, the occurrence relating to a first component;

compare the first component along with other components to at least one fingerprint, wherein the fingerprint represents key low-level elements of a model of a known component; and

if the first component and the other discovered components match substantially all of the key low-level elements of the fingerprint, use a subfingerprint of a known refined component to discover the existence of a second component that corresponds to the known refined component.

25. A method for managing components in an information technology (IT) system, comprising:

receiving a first event message for a first occurrence in the IT system, the first occurrence relating to a first component;

if the first component matches at least one low-level element of a fingerprint of a model of a known component, adding the first component to an accumulator;

if all of the low-level elements of the fingerprint have been matched by the first component and other components, generating a command to detect further information;

receiving, in response to the command, a second event message providing further details about one of the components; and

using a subfingerprint of a known refined component and the further details about one of the components to discover a refined component.

26. The method of claim 25, wherein the first occurrence is one of a file creation, file deletion, file modification, registry key creation, registry key modification, and registry key deletion.

27. The method of claim 25, further comprising:

generating a component detected message upon the discovery of the refined component;

retrieving a list of elements to track for the refined component; and

using the list of elements to track changes to the refined component.

28. A method for discovery of a refined component in an information technology (IT) system, comprising:

using a fingerprint of a model of a known component to discover an existing component in the IT system by matching passive elements in the fingerprint with event information of the IT system;

generating and transmitting a command message defined by active elements of the fingerprint to discover the refined component;

receiving event information relating to the active elements of the fingerprint of the known component; and

using a subfingerprint of the refined component to discover the refined component, the refined component relating to the known component, wherein the subfingerprint of the refined component becomes active upon the discovery of the existing component using the fingerprint.

29. The method of claim 28, wherein receiving event information relating to active elements includes receiving an event message.

30. A method for determining dependencies between at least two components in an information technology (IT) system, comprising:

discovering the at least two components in the IT system;

monitoring the usage of resources by the two components in the IT system and, if a resource is used by one of the two components, generating a message indicating the use of that resource by that component;

accumulating each message indicating the use of one of the resources by one of the two components; and

if the accumulated messages indicate that the two components use the same resource, then indicating that a dependency between the two components has been detected.

31. The method of claim 30, further comprising determining a direction of the dependency between the two components.

32. The method of claim 30, wherein the component is selected from the group consisting of an application, a network connection endpoint, and a server.

33. The method of claim 32, wherein at least one message indicates a network outbound connection by one of the two components.

34. The method of claim 32, wherein at least one message indicates a network listener by one of the two components.

35. The method of claim 32, wherein at least one message indicates a use of a file by one of the two components.

36. The method of claim 30, further comprising tracking changes to the dependency between the two components.

37. The method of claim 30, wherein the dependency is a containment dependency.

38. The method of claim 30, wherein the dependency is a network dependency.

39. The method of claim 30, wherein the dependency is a shared usage dependency.

40. An apparatus for determining dependencies between at least two components in an information technology (IT) system, comprising:

a memory storing a program;

a processor in communication with the memory; in which the processor is directed by the program to:

discover the at least two components in the IT system;

monitor the usage of resources by the two components in the IT system and, if a resource is used by one of the two components, generating a message indicating the use of that resource by that component;

accumulate each message indicating the use of one of the resources by one of the two components; and

if the accumulated messages indicate that the two components use the same resource, then indicate that a dependency between the two components has been detected.

41. A method for tracking content changes to a component in an information technology (IT) system, comprising:

generating an event message for an occurrence in the IT system, the occurrence relating to the component;

if contents are to be tracked for the component, comparing current contents of the component with a previous version of the contents of the component; and

logging differences between the current contents of the component and the previous version of contents of the component.

42. The method of claim 41, further comprising:

generating a command to copy the current contents of the component; and

in response to the command, receiving the current contents of the component.

43. An apparatus for tracking content changes to a component in an information technology (IT) system, comprising:

a memory storing a program;

a processor in communication with the memory; in which the processor is directed by the program to:

generate an event message for an occurrence in the IT system, the occurrence relating to the component;

if contents are to be tracked for the component, compare current contents of the component with a previous version of the contents of the component; and

log differences between the current contents of the component and the previous version of contents of the component.

44. A system for collecting information on components in an information technology (IT) system, comprising:

a plurality of agents, wherein each agent resides on a computer of the IT system, and wherein each agent includes instructions to: (a) discover components in the IT system, (b) determine at least one dependency between two or more of the discovered components, and (c) track changes to the discovered components and the dependency between two or more of the discovered components; and

a network server in communication with the plurality of agents, wherein the network server includes instructions to receive component detection messages from the agents and generate a visual map of the discovered components.